GENERAL SOFTWARE QUALITY ASSURANCE

PROCEDURE ID: YMP-LBNL-QIP-SI.0 | REV. 2, MOD. 0 | EFFECTIVE: 6~~1~~/2~~17~~/97

1. PURPOSE

This quality implementing procedure (QIP) establishes a quality assurance (QA) methodology for controlling and documenting the development, modification, verification, validation, distribution, use, and withdrawal/retirement of software by the Yucca Mountain Project—~~Lawrence~~Ernest Orlando Lawrence Berkeley National Laboratory (YMP-LBNL).

2. SCOPE

This procedure applies to quality-affecting software developed, modified or acquired by YMP-LBNL, including commercial-grade software. Software within the scope of this QIP is authorized for continued YMP-LBNL use provided the YMP-LBNL Software Configuration Management system includes documentation that the software has a current released status. Software without a current released status requires completion of additional documentation, as indicated by the SCM system, prior to release for authorized use.

Software that is acquired as an integral part of measuring and test equipment is controlled by YMP-LBNL-QIP-12.0 and is exempt from this QIP. The following types of software do not fall within the scope of this QIP:

- operating systems, administrative and management systems, and system utilities, and compilers and their associated libraries;
- software that does not generate any data, such as word processing programs, spreadsheets, database managers, graphing and visual display software, plotting routines, and statistical analysis programs. However, applications written for use within these types of software may need to meet the requirements for the software covered by this procedure.

3. PROCEDURE

This section presents requirements for the three distinct categories of software within the scope of this procedure: developed or modified software, acquired software, and software verifiable by visual inspection and/or hand calculations. The YMP-LBNL staff shall identify the category of the software under review, and proceed with the activities shown in the section(s) pertaining to that category.

3.1 Determination of Software Life Cycles, Baselines, and Controls

The YMP-LBNL shall perform software life cycle (or other) activities for the software under review, per the requirements of the appropriate following section.

3.1.1 Software Life Cycle Plan for Developed or Modified Software

The YMP-LBNL staff shall perform software life cycle activities, in an iterative or sequential manner. This section constitutes the Software Life Cycle Plan for all developed or modified software used in YMP-LBNL activities, unless a specific Software Life Cycle Plan is made for the software, and approved by the Software ~~Q~~EA Specialist or designee and Project Manager or designee. The outline of the Software Life Cycle Plan and associated control points is shown schematically in Figure 1. Each of the following paragraphs describes a phase of the Software Life Cycle Plan, and defines any related control point(s) for that phase.

A. Prototyping. This phase is exploratory and may be carried out in a flexible, interactive, and iterative manner. Work shall be recorded in the scientific notebook according to the requirements of YMP-LBNL-QIP-SIII.0~~, "Scientific Investigation"~~. Results from this phase shall be used as input for the required documentation and reviews of the following phases (see also Section 3.4 of this procedure). The software control point for the completion of this phase is the submission of a Software Identification Form (Attachment 1 to this procedure, plus additional attachments as needed) to the Software ~~Q~~EA Specialist, indicating that the software is ready to proceed through the subsequent life cycle phase. The results of this phase shall be used as input to the software baseline documents described in Section 3.3 of this procedure, although further work may be done if additional results are necessary.

B. Requirements Specification. This phase defines the task or problem to be addressed by the software. The Software Configuration Management Form (Attachment 2 to this procedure, plus additional attachments as needed) for this software is begun in this phase. The software control point for the completion of this phase is the approval of the software verification of the Requirements Specification defined in Section 3.4.2.

C. Design Description. This phase consists of developing the formal structure and architecture of the software (for example, a flow chart). The software control point for the completion of this phase is the approval of the software verification of the Design Description defined in Section 3.4.3.

D. Implementation and Software Validation. This phase includes the implementation of the design in computer code and the testing, debugging, and validating of the resulting software. The software control points for the completion of this phase are the approval of the Software User Documentation and approval of the Software Validation Report defined in Sections 3.4.4 and 3.4.5, respectively.

Note: Software validation is not to be confused with model validation in the scientific literature.

E. Operation and Maintenance. This phase consists of the use of the software to support YMP-LBNL activities. This phase may also include modification, software defect identification and resolution, distribution, and withdrawal or retirement of the software as defined in Section 3.6

3.1.2 Software Life Cycle Plan for Acquired Software

The YMP-LBNL staff shall perform software life cycle activities on acquired software, as described in Section 3.1.1 of this procedure, except for the omission of the Prototyping and Design phases. All other phases and related control point(s) for each phase are performed as described in Section 3.1.1.

Additionally, the YMP-LBNL staff shall perform installation tests on acquired software, to ensure that the software performs as required in the operational environment. The staff shall complete the Software Identification Form (Attachment 1 to this procedure, plus additional attachments as needed) and submit it to the Software ~~Q~~EA Specialist.

3.1.3 Requirements for Software Verifiable by Visual Inspection and/or Hand Calculations.

No life cycle activities are required for this software category. The YMP-LBNL staff shall list the baseline version and subsequent changes to the software in project documentation, and also shall document that the software provides correct results for a specified range of input parameters.

3.2 Software Verification Plan

The YMP-LBNL staff shall perform software verification activities as defined by a Software Verification Plan. In lieu of a specific plan for each software item, this section shall constitute the Software Verification Plan for the software used in the YMP-LBNL activities, and consists of the following points:

A. Software verification shall be accomplished by reviewing each of the software baseline documents (see Section 3.3 of this procedure) against the corresponding criteria specified in this procedure.

B. The review shall be performed according to YMP-LBNL-QIP-6.1~~, *“Document Review”*~~. Reviewers shall be selected according to YMP-LBNL-QIP-6.1 except that each reviewer shall be independent of the development or modification of the software to be reviewed. The person who directed the work may perform these activities with a higher level of management approval and documented justification.

C. The verification documentation shall include a description of the tasks, methods, implementing documents, and acceptance criteria, as appropriate, for accomplishing the software verification. It shall also document the results of the software verification, including the extent to which the results agree with the specified acceptance criteria listed for each software baseline document.

Note: Software verification is simply a review procedure in software quality assurance and is not to be confused with code verification (e.g., verification of coded mathematical equations) in the scientific literature.

3.3 Documentation Requirements

Software shall be documented sufficiently by the YMP-LBNL staff to allow an independent repetition of the use of the software and to demonstrate the ability of the software to meet the requirements of this QIP.

For each software category, the documents listed below constitute the software baseline documents for the software. Such documents shall clearly identify any of the requirements shown here that are not applicable for the document. The review and approval process for each baseline document is described in Section 3.4 of this procedure.

3.3.1 Documentation for Developed or Modified Software

Required documentation for developed or modified software shall consist of, as applicable:

- Software Identification Form (Attachment 1 to this procedure, plus additional attachments as needed)
- Requirements Specification - a description of the overall nature and purpose of the software, and requirements for its intended use, including general acceptance criteria as appropriate
- Design Description-technical information, including:
  -performance requirements and design constraints
  -interfaces with external data, hardware, or other software
  -applicable software and hardware operation issues including programming languages and versions, portability, maintainability, reliability, and efficiency
  -a description of each software item as it relates to the functional requirements specification
  -a description of the software structure including software internal interfaces, control logic, and data structure and flow
  -a description of models and numerical methods
  -source code, either electronically or printed
- Software User Documentation - a description of how to use the software, including:
  -input and output options
  -data files, input and output data, defaults and file formats
  -a description of the allowable and tolerable ranges for inputs and outputs
  -anticipated errors and how the user can respond
  -hardware and software operating environments
  -available sample problems
  -installation procedures
- Software Validation Report - to include:
  -documentation of test methods and test cases, showing that the software meets the requirements for its intended use.
  -a description of tasks, methods, implementing documents, and acceptance criteria used
  -a record of the results of the validation, including the extent to which the results agree with the specified acceptance criteria

3.3.2 Documentation for Acquired Software

Required documentation for acquired software shall consist of, as applicable:

- Software Identification Form (Attachment 1 to this procedure, plus additional attachments as needed)
- Requirements Specification - a description of the overall nature and purpose of the software, and requirements for its intended use, including general acceptance criteria as appropriate
- Design Description - technical information, including:
  -interfaces with external data, hardware or other software
  - description of models and numerical methods
- Software User Documentation - a description of how to use the software, including:
  - input and output options
  - data files, input and output data, defaults, and file formats
  - a description of the allowable and tolerable ranges for inputs and outputs
  - anticipated errors and how the user can respond
  - hardware and software operating environments
  - available sample problems
  - installation procedures
- Software Validation Report - to include:
  - documentation of test methods and test cases, showing that the software meets the requirements for its intended use, using test cases developed independently of the software developer. Additional test cases provided by the developer may be used to supplement this process with justification for their use.
  - a description of tasks, methods, implementing documents, and acceptance criteria used
  - a record of the results of the validation, including the extent to which the results agree with the specified acceptance criteria

3.3.3 Documentation for Software Verifiable by Visual Inspection and/or Hand Calculations

Required documentation for software verifiable by visual inspection and/or hand calculations shall consist of as applicable:

- A listing of the baseline version and subsequent changes to the software
- Documentation that the software provides correct results for a specified range of input parameters

Note: Software acquired from outside YMP-LBNL or developed prior to implementation of the YMP-LBNL QA Program may not have been documented according to the YMP-LBNL Software Life Cycle Plan. For developed, modified, and/or acquired software of this status, software validation shall be performed and verified against criteria documented in the Requirements Specification, installation testing shall be conducted to ensure that the software performs as required in the operational environment, and software user documentation will be made available. Clarification shall be provided with the Software Configuration Management Form (Attachment 2 to this procedure, plus additional attachments as needed) for any unavailable documentation.

3.4 Documentation Completion, Review and Approval

The YMP-LBNL staff shall complete each software baseline document, as identified in section 3.3 for the category of the software under review, per the instructions below. The Software ~~Q~~EA Specialist or designee shall approve the documents following satisfactory completion of a YMP-LBNL-QIP-6.1 review for documents either singly or as part of a larger report including resolution of review comments, per the Software Verification Plan.

3.4.1 Software Identification Form. At the completion of the prototyping phase, the Software Identification Form (Attachment 1, plus additional attachments as needed) shall be completed by the Technical Contact and signed by the PI. This form shall be submitted to the ~~Quality Assurance Manager~~Software EA Specialist or designee after completion and sign-off before the software may be used for quality-affecting work.

Upon receipt of the Software Identification Form, the Software ~~Q~~EA Specialist or designee, in consultation with the Technical Contact, shall assign a unique identifier to the software. This identifier shall be used to identify all associated software documents and to provide a cross-reference to other software baseline documents in effect. Each version of the software shall have a unique identifier. It shall also be listed on all outputs, if feasible.

A. For acquired software, the Software Identification Form shall provide name(s) and date(s) of the creation or last modification of the executable software. If multiple files are required, the list shall indicate the name of the file used to start the software.

B. For developed software, the equivalent information shall be provided with the machine-readable copy of the source and/or executable code which is submitted in accordance with Paragraph

~~3.5.3~~ 3.5.3 of this procedure. Once documented, this information is used to maintain configuration control of the identified software.

3.4.2 Requirements Specification. The Software Configuration Management Form (Attachment 2, plus additional attachments as needed) for this software is begun to be filled in during the development of this document.

Software verification of the Requirements Specification shall be performed by review. The review shall verify that the document is complete. In addition, the review shall ensure that the software developed or selected has been evaluated to be applicable to the problem being solved and that the specified requirements can be validated.

3.4.3 Design Description. Software verification of the Design Description shall be performed by review. The review will verify that the document is complete. In addition, the review shall confirm that the requirements are reflected in the design.

3.4.4 Software User Documentation. Software verification of the Software User Documentation shall be performed by review. The review will verify that the document is complete and that inputs are within the appropriate range. It will also verify that assumptions are valid and traceable, explicitly stated, and that the assumptions are not against basic physical principles.

3.4.5 Software Validation Test Plan and Report. Software verification of the Software Validation Report shall be performed by review. The review will verify that the document is complete. In addition, the review shall confirm that the software validation plan and software validation are adequate for the use(s) of the software.

3.5 Release of Approved Software

When all required documents have completed their software verification reviews, the YMP-LBNL Software ~~Q~~EA Specialist or designee shall verify that all QIP requirements have been met and shall approve release of the software for YMP use, with the approval of the Project Manager (or designee). The YMP-LBNL Software ~~Q~~EA Specialist or designee shall notify the Technical Contact that the software has been released.

3.5.1 After release of the software a YMP user shall use the software as
stated ~~stated~~ on the Requirements Specification.

3.5.2 In those cases where work is performed with software prior to the software's ~~Q~~EA approval, the portions of the software that have not been verified and validated shall be identified and controlled, and written justification of the reason shall be documented.

3.5.3 A machine-readable copy of the source code and executable code shall be identified as the baseline copy by the Technical Contact, if available. If source code is not provided, a document stating the justification shall be provided instead.

3.6 Software Configuration Management

Configuration identification, configuration control, and status accounting shall be maintained in the Software Configuration Management System in accordance with the following sections. Software shall be placed under configuration management control as each baseline element is approved. A diagram of the operation and maintenance of software in the Software Configuration Management System is provided in Figure 2.

3.6.1 Software Configuration Status Accounting. The YMP-LBNL Records Staff shall initiate and maintain a Software Configuration Status Accounting Log which shall contain information for all identified software.

A. The log shall record

1. a listing of all software baseline documents, software configuration documents, and associated unique identifiers;

2. all documented YMP software users;

3. the status and brief descriptions of all documented software problems; and

4. the status and a brief description of proposed and approved software changes.

B. The log shall also record characteristics of the software file(s) adequate to identify the managed copies of the software. These characteristics are the name(s) and date(s) of the creation or last modification of the executable software, and where to access the files.

C. The Software ~~Q~~EA Specialist or designee shall provide a Software Configuration Status Accounting Log Update to the YMP-LBNL Records Processing Center on a quarterly basis. The Software Configuration Status Accounting Log Update shall contain all changes that were recorded within the preceding three months. The Software Configuration Status Accounting Log shall be submitted annually to the LBNL Records Processing Center.

3.6.2 Software Distribution. Released software may be used by other YMP participants provided that the software is appropriate for the application and the use is within the requirements as stated in the Requirements Specification. Use by non-YMP users shall not be regulated.

3.6.3 Software Change Control. Any change to released software and/or software baseline document(s) shall be proposed by providing the Technical Contact with updated software baseline document(s), which shall include a description of the change, the rationale for the change, and the identification of affected baseline document(s). A Software Configuration Management Form (Attachment 2, plus additional attachments as needed) and a new Software Identification Form (Attachment 1, plus additional attachments as needed) shall accompany the updated software baseline document(s).

A. Software verifications shall be performed for the changes as necessary to ensure that the change is appropriately reflected in software documentation and to ensure that document traceability is maintained.

B. Also, software validation shall be performed as necessary for the change, including regression testing. Changes that do not affect the functionality of the code, but that make merely cosmetic or editorial modifications, are exempted from formal change control, but are under revision control.

C. All of these document(s) and tests shall be subject to the review and approval process and recorded on the appropriate spaces on the Software Configuration Management Form (Attachment 2, plus additional attachments as needed).

D. Written information concerning changes shall be transmitted to all YMP organizations affected by the changes by the YMP-LBNL Records Staff.

3.6.4 Software Defect Identification and Resolution. A suspected software defect found by any YMP user shall be reported on a Software Configuration Management Form (Attachment 2, plus additional attachments as needed) to the Software ~~Q~~EA Specialist, who shall have it recorded in the Software Configuration Management Status Accounting Log, and then forward it to the Technical Contact for a determination of whether there is an actual defect.

A. If the presence of a defect is confirmed, the Technical Contact shall sign and send the Software Configuration Management Form (Attachment 2, plus additional attachments as needed) to the Software ~~Q~~EA Specialist for tracking.

1. Identified defects shall be reported to any users of the software by the YMP-LBNL Records Staff.

2. A Software Defect Resolution Report shall be prepared by the Technical Contact. This report shall include proposed modifications to the software and/or the software baseline documents and shall discuss potential impacts of the defect, if any, on previous applications.

3. The Software Defect Resolution Report shall be reviewed in accordance with YMP-LBNL-QIP-6.1 and shall be approved by the Project Manager or designee on the Software Configuration Management Form (Attachment 2, plus additional attachments as needed).

4. Modification to the software baseline documents, if any, shall then be performed in accordance with Section 3.6.3.

5. The YMP-LBNL Records Staff shall disseminate a copy of the resolution report to all users of the software. If a defect is identified that adversely impacts previous applications by the user, the condition adverse to quality shall be documented by the user and controlled in accordance with AP-16.1Q and AP-16.2 Q.

B. If it is not a defect, an explanation of the occurrence shall be written by the Technical Contact and attached to the Software Configuration Management Form. The Software ~~Q~~EA Specialist or the Technical Contact or a designee shall communicate the explanation to the one submitting the report.

3.6.5 Software Retirement or Withdrawal. Retirement or withdrawal of released software shall be documented by submitting a completed Software Configuration Management Form (Attachment 2, plus additional attachments as needed) to the Software ~~Q~~EA Specialist by the Principal Investigator or Technical Contact holding responsibility for the software.

A. If a Technical Contact wishes to withdraw or retire software, the Project Manager or designee shall approve the withdrawal/ retirement on the Software Configuration Management Form (Attachment 2, plus additional attachments as needed).

B. Information concerning retirements or withdrawals shall be transmitted to all affected software users by the YMP-LBNL Records Staff.

C. If another Principal Investigator or Technical Contact wishes to continue use of the software, the new Principal Investigator will transfer responsibility for the software to the new Technical Contact by submitting a new Software Identification Form (Attachment 1, plus additional attachments as needed).

3.7 Software Application and Use

3.7.1 If a user is within YMP-LBNL, then the Principal Investigator or designee for the additional software user shall identify the application of the software on a Software Configuration Management Form (Attachment 2, plus additional attachments as needed). The signature of an independent reviewer on the Software Configuration Management Form shall be provided to ensure that the software selected is applicable to the problem to be solved and that inputs and assumptions are valid and traceable, so that comparable results can be obtained, with any differences explained, through independent replication of the process.

3.7.2 For other YMP participants, written justification of the suitability of the software for their application shall be requested by and kept on file with the YMP-LBNL Records Processing Center.

3.7.3 If the use of a software item falls outside the existing range of validation, further validation shall be performed prior to use by the user.

4. RECORDS MANAGEMENT

4.1 Lifetime

Completed Software Identification Forms, plus any attachments (Attachment 1)
Completed Software Configuration Management Forms, plus any attachments (Attachment 2)
Exhibits and notifications as required by the Attachments
Software Configuration Status Accounting Log (annual)
Magnetic or other storage media for software developed under this procedure

4.2 Non-permanent

None.

4.3 Controlled Documents

None.

4.4 Records Center Documents

Records associated with this procedure shall be submitted to the YMP-LBNL Local Records Processing Center, in accordance with YMP-LBNL-QIP-17.0 and AP-17.1Q.

5. RESPONSIBILITIES

5.1 The **Project Manager** has programmatic responsibility for implementation of this QIP. The Project Manager is also responsible for assuring the implementation of this QIP by personnel engaged in YMP-LBNL scientific investigations, studies, and activities. The Project Manager or designee is also responsible for approval of the software baseline documents, for software defect resolution reports, and for software release and withdrawal.

~~5.2 **Quality Assurance Manager** is responsible for assuring the software quality assurance process is performed as required.~~

5.~~3~~2 The **YMP-LBNL Software ~~Quality~~Engineering Assurance Specialist** is responsible for assuring that all software documented by this QIP satisfies the requirements set forth herein prior to release of the software, and oversees the Software Configuration Management System.

5.~~4~~3 The **Principal Investigators (PIs)** are responsible for assigning Technical Contacts for software, and assuring the preparation of software documents appropriate for their activities.

5.~~5~~4 The **Technical Contact** has the immediate responsibility for implementing the provisions of this QIP for the YMP-LBNL software that is assigned to the Technical Contact.

5.~~6~~5 The **YMP- LBNL Records Staff** is responsible for maintaining the Software Configuration Status Log and for notifying software users of any changes, defects, resolutions of defects, and withdrawals of software.

6. ACRONYMS AND DEFINITIONS

6.1 Acronyms

SCM Software Configuration Management

6.2 Definitions

Numerical Methods: Methods for obtaining approximations for solutions of mathematical equations that cannot be solved exactly. Numerical methods involve the substitution of simpler equations that are then solved exactly and/or yield solutions with errors which are not quantifiable. Examples of numerical methods include numerical integration and differentiation (e.g., finite difference or finite element methods).

Revision: An original software document or changes to an approved software document.

Site Characterization Plan (SCP) Activity Number: Unique identification of discrete segments of YMP project activities.

Software: Computer operations specified in any language(s) that function as a single unit.

Software Baseline Document(s): A document, a set of documents, or a product formally designated and controlled at a specific time during the software life cycle.

Software Configuration Management: A system of controls and authorizations which prevents ambiguity as to which version of software is used for a particular computation.

Software Configuration Status Accounting: The recording and reporting of information that is necessary to manage the development, modification, and operation of software.

Software Control Point: Milestones in the software life cycle at which controls are applied to the software.

Software Defect: An error in a computer code (synonymous with "bug") or a failure to meet one or more specifications of the software.

Software Life Cycle: A series of activities that begin when software is conceived and ends when the software is no longer available for use.

Software ~~Q~~EA Specialist: A YMP-LBNL staff member appointed by the YMP-LBNL ~~Quality Assurance~~Project Manger and responsible for LBNL's software quality assurance.

Software Validation: The process of evaluating a software system or component during or at the end of the development process to determine whether it satisfies specified requirements.

Software Verification: The process of determining whether the products of a given software life cycle phase satisfy the conditions imposed at the start of that phase.

Version: Designation of the original software or changes to released software, usually numeric.

The OQA Representative is responsible for review of software for compliance with applicable QARD requirements.

7. REFERENCES

- AP-16.1Q, *Performance/Deficiency Reporting*
- AP-16.2Q, *Corrective Action and Stop Work*
- *AP-17.1Q, Record Source Responsibilities for Inclusionary Records*
- DOE/RW/0333P, *Quality Assurance Requirements and Description (QARD)*
- YMP-LBNL-QIP-6.1, *Document Review*
- YMP-LBNL-QIP-12.0, *Documenting the Usage of Measuring and Test Equipment*
- *YMP-LBNL-QIP-SIII.0, Scientific Investigation*
- YMP-LBNL-QIP-17.0 *Submitting Records to the YMP-LBNL Records Processing Center*

8. ATTACHMENTS

Figure 1.	Software Life Cycle Phases and Associated Documents and Reviews
Figure 2.	Software Configuration Management for Software Operation and Maintenance
Attachment 1.	Software Identification Form
Attachment 2.	Software Configuration Management Form

9. REVISION HISTORY

_9/6/96 - Revision 0, Modification 1:
Revised procedure for handling of Software Configuration Accounting Status Log

10/22/96 - Revision 1, Modification 0:
Revised procedure to reflect requirements changes in QARD, Rev. 5.

_1/10/97 - Revision 1, Modification 1:
Deleted requirements in Sections 3.4 and 3.5 for individual document approvals by QA Specialist or Designee. Modified Software Config. Mgmt. Form Attach.2.

6/2/97 -Revision 2, Modification 0:
Revised procedure to introduce the term Engineering Assurance (EA), changed Software QA Specialist to Software EA Specialist, and deleted QA Manager.

10. APPROVALS

Preparer: Date

Technical Reviewer: Date

Technical Reviewer: Date

QA Reviewer: Date

Q~~uality ~~A~~ssurance~~ Manager: Date

Project Manager: Date